UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/789,028	02/25/2004	Michael Gerard Hinchey	GSC 14, 389-1	7158

21872 7590 05/27/2008

NASA GODDARD SPACE FLIGHT CENTER
8800 GREENBELT ROAD, MAIL CODE 140.1
GREENBELT, MD 20771

EXAMINER	
DENG, ANNA CHEN	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
05/27/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/789,028	HINCHEY ET AL.	
	Examiner	Art Unit	
	ANNA DENG	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 February 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *11/4/2007 3/4/2008*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to amendment filed on 2/4/2008.

2. Claims 1-28 are pending.

Response to Amendment

3. The rejection under U.S. C. 101 to claims 26-27 is withdrawn in view of applicant's amendment.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-4, 10-23, and 26-28 are rejected under U.S.C. 35 102 (b) as being anticipated by Garland et al., US 6,289,502 B1 (hereinafter Garland).

Per Claim 1 (Currently Amended):

Garland discloses:

- A method for deriving a process-based specification for a system (Garland, col. 2, lines 25-36, "The method includes accepting, a design specification for the distributed system, including accepting specifications of multiple state machines, and accepting a specification of desired properties of the state machines"), comprising:

- deriving a trace-based specification from a non-empty set of traces (Garland, col. 2, lines 46-58, "each specification of a state transition includes a specification of values of the state variables in a state in which the transition can be taken, and a specification of an effect on the values of the state variables when the transition is taken"; col. 3, lines 10-15, deriving a

second design specification from the accepted design specification, wherein the derived design specification has a property (non-empty set); col. 7, lines 29-34, “a new derived specification in the IOA language then becomes part of the user specification”);also, col. 22, lines 4-11, "After inputting an initial abstract specification, and after each iteration in which a new specification is derived, the user can determine whether the system specification has the required invariant properties specified by the user…"); and

- mathematically inferring the process-based specification from the trace-based specification, wherein the process-based specification is mathematically equivalent to the trace-based specification (Garland, col. 2, lines 7-15, “The present invention uses a new computer language which is based on a formal, mathematical state-machine model, and which is used both to validate and to generate code for a distributed system. In general, use of this new language enables developing a system using multiple related system specification”, FIG. 7, col. 13, lines 58-65, “Automaton preprocessor 710 takes the specification of automaton A630, which can make full use of the features of the IOA language, into a from that is usable by the theorem-prover. This form is a mathematical description of the underlying automaton, giving its actions, states (and start states), transitions, and tasks explicitly, in basic logical notation” (emphases added)).

Per Claim 2:

Garland discloses:

- the process-based specification is provably equivalent to the trace-based specification (col. 9, lines 14, “a preprocessor that accepts a specification in the IOA language and produces an equivalent specification in the language of the core tool”).

Per Claim 3 (Currently Amended):

Garland discloses:

- generating the process-based specification using an inference engine (theorem prover) (Garland, col. 2, lines 29-35, "including applying a theorem proving procedure to the design specification, and also includes applying a code generating procedure to the specifications of the state machines to generate multiple software implementation for components of the distributed system"; col. 2, lines 62-67, "the step of applying the theorem proving procedure to the design specification can include translating the design specification into a logical language of a theorem prover and providing the translated design specification to a theorem prover, such as a software based theorem proving program (emphasis added)").

Per Claim 4 (Currently Amended):

Garland discloses:

- the laws of concurrency are used by the inference engine to generate the process-based specification (Garland, FIGS. 11-12, col. 18, lines 35-67, "searching techniques can make use of "symmetries" in the states of an automaton to concurrently test a predicate in multiple states of the automaton").

Per Claim 10 (Currently Amended):

Garland discloses:

- analyzing the process-based specification to examine possible implementations of the process-based specification in different configurations (col. 2, lines 10-35, "developing a system using multiple related system specification… The method includes accepting a design specification for the distributed system, including accepting a design specification of multiple state machines…")

Per Claim 11 (Currently Amended):

Garland discloses:

- the various possible implementations of the process-based specification are based on transformations of the process-based specification by application of the laws of concurrency to derive various implementations (col. 2, lines 29-58, “the state machines have the desired properties, including applying a theorem proving procedure to the design specification, and also includes applying a code generating procedure to the specification of the state machines to generate multiple software implementations for components of the distributed system”).

Per Claim 12:

Garland discloses:

- the various equivalent implementations are mathematically equivalent to the process-based specification (col. 2, lines 62-67, “applying the theorem proving procedure to the design specification can include translating the design specification into a logical language of a theorem prover and providing the translated design specification to a theorem prover, …The theorem prover can be an equational theorem prover”).

Per Claim 13:

Garland discloses:

- the various equivalent implementations are provable equivalent to the process-based specification (col. 2, lines 29-58, “the state machines have the desired properties, including applying a theorem proving procedure to the design specification, and also includes applying a code generating procedure to the specification of the state machines to generate multiple software implementations for components of the distributed system”) .

Per Claim 14:

Garland discloses:

- multiple correct process-based specifications are possible (col. 2, lines 8-36, "both to validate and to generate code for a distributed system…developing a system using multiple related system specifications…).

Per Claim 15 (Currently Amended):

Garland discloses:

- deciding which of the multiple correct process-based specifications are most appropriate (col. 2, lines 46-61, "The specification of each state machine can include a specification of a multiple state variables, the values of which determine the state of the state machine…").

Per Claim 16:

Garland discloses:

- the process-based specification is used as a basis for generation of alternate representations (col. 11, lines 19-26).

Per Claim 17:

Garland discloses:

- the alternate representations are sets of instructions (col. 11, lines 19-26).

Per Claim 18:

Garland discloses:

- the set of traces is a set of sequences of events or activities specific to an application domain (col. 3, lines 1-16).

Per Claim 19:

Garland discloses:

- the set of traces is derived by pre-processing a set of scenarios given as input by a user to a context sensitive editor (FIG. 1A, col. 6, lines 1-46).

Per Claim 20:

Garland discloses:

- the set of scenarios is natural language text describing intended system behavior, and the elements of the set of traces are sequences of events or activities in a given application domain (col. 7, lines 14-23, "user specification 100 includes a set of text files, … Each text file provides a portion of user specification 100, for example, the specification of one component of the system. The user edits these text files using a standard text editing program. After creating or modifying one or more of these specification text files, the user can use a development tool to process the text files.").

Per Claim 21:

Garland discloses:

- the set of scenarios is represented by various graphical notations (FIG. 1A, col. 6, lines 32-35, "a user interface 63, such as a graphical display").

Per Claim 22:

Garland discloses:

- the deriving step is repeated (col. 16, lines 29-39, "Simulator 910 performs a loop where, in each iteration, it uses the user-provided function to determine the next transition definition and parameter values and then executes the effect part of that transition definition with those parameter values").

Per Claim 23:

Garland discloses:

- the inferring step is repeated (col. 16, lines 29-39).

Per Claim 26 (Currently Amended):

This is the system version of the claimed method discussed above (claim 1) where in all claim limitations also have been addressed and /or covered in cited areas as set forth above, including at least one natural language scenario; a trace-based specification derived from the at least one natural language scenario (Garland, col. 7, lines 14-23, "user specification 100 includes a set of text files, … Each text file provides a portion of user specification 100, for example, the specification of one component of the system. The user edits these text files using a standard text editing program. After creating or modifying one or more of these specification text files, the user can use a development tool to process the text files.") Thus, accordingly, this claim is also anticipated by Garland.

Per Claim 27 (Currently Amended):

This is the system version of the claimed method discussed above (claim 1) where in all claim limitations also have been addressed and /or covered in cited areas as set forth above. Thus, accordingly, this claim is also anticipated by Garland.

Per Claim 28 (Currently Amended):

This is another version of the claimed method discussed above (claim 1) where in all claim limitations also have been addressed and /or covered in cited areas as set forth above, including receiving at least natural language scenario describing the action; generating a trace-based specification from the at least one natural language scenario (Garland, col. 7, lines 14-23, "user specification 100 includes a set of text files, … Each text file provides a portion of user specification 100, for example, the specification of one component of the system. The user edits these text files using a standard text editing program. After creating or modifying one or more of these specification text files, the user can use a development tool to process the text files.") Thus, accordingly, this claim is also anticipated by Garland.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 5-9, and 24-25 rejected under 35 U.S.C. 103(a) as being unpatentable over Garland et al. US 6,289,502 B1 (hereinafter Garland), in view of Bowman-Amuah US 6,405,364 B1 (hereinafter Bowman-Amuah).

Per Claim 5:

Garland teaches laws of concurrency (Garland, FIGS. 11-12, col. 18, lines 35-67, "searching techniques can make use of "symmetries" in the states of an automaton to concurrently test a predicate in multiple states of the automaton"); Garland does not explicitly teach are reversed by embedding the laws of concurrency in the inference engine. However, Bowman-Amuah teaches are reversed by embedding the laws of concurrency in the inference engine (Bowman-Amuah, col. 39, lines 23-40, "… "round-trip" reengineering" provides the developer with a way of modifying a component model and generating the code, then at a later date modifying the code at predefined locations in the source code and regenerating, thus enabling the model to maintain a 2-way-syncronization").

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Garland to include "are reversed by embedding the laws of concurrency in the inference engine" using the teaching of Bowman-Amuah. The modification would be obvious because one of ordinary skill in the art would be motivated to build systems in a development architecture framework to fulfill the requirement (Bowman-Amuah, col. 2, lines 18-23).

Per Claim 6:

The rejection of claim 5 is incorporated, and Garland further teaches the embedding is syntactic or shallow (Garland, col. 11, lines 5-7, "indicated syntactically using a prime after a variable name to signify a value in a pre-state").

Per Claim 7:

The rejection of claim 5 is incorporated, and Garland further teaches the embedding is semantic or deep (Garland, col. 13, lines 12-15, "a static semantic checker can determine…").

Per Claim 8:

Garland teaches laws of concurrency (Garland, FIGS. 11-12, col. 18, lines 35-67, "searching techniques can make use of "symmetries" in the states of an automaton to concurrently test a predicate in multiple states of the automaton"); output in response to a given input of at least one trace (Garland, col. 7, lines 25-30, inputs specification text files and produces an output such as text message); Garland does not explicitly teach are reversed so that an equivalent process expression is output in response to a given input at least one trace. However Bowman-Amuah teaches are reversed so that an equivalent process expression is output in response to a given input of at least one trace (Bowman-Amuah, col. 39, lines 23-40, "… "round-trip" reengineering" provides the developer with a way of modifying a component model and generating the code, then at a later date modifying the code at predefined locations in the source code and regenerating, thus enabling the model to maintain a 2-way-syncronization").

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Garland to include "are reversed so that an equivalent process expression is output in response to a given input of at least one trace" using the teaching of Bowman-Amuah. The modification would be obvious because one of ordinary skill in the art would be motivated to build systems in a development architecture framework to fulfill the requirement (Bowman-Amuah, col. 2, lines 18-23).

Per Claim 9:

The rejection of claim 5 is incorporated, and Garland further teaches multiple process expressions are given as output in response to inputs of the at least one trace (FIGS. 6-7, lines 20-56, "Invariant theorem prover 610 takes as input a specification of an I/O automaton A 630, … Validation output includes theorem prover output 660.Theorem prover output 660 can include a declaration that invariant I_A 640…").

Per Claim 24 (Currently Amended):

Garland teaches using the deriving step and the inferring step (Garland, col. 2, lines 46-58, "each specification of a state transition includes a specification of values of the state variables in a state in which the transition can be taken, and a specification of an effect on the values of the state variables when the transition is taken"; col. 13, lines 58-65, "Automaton preprocessor 710 takes the specification of automaton A630, which can make full use of the features of the IOA language, into a from that is usable by the theorem-prover. This form is a mathematical description of the underlying automaton, giving its actions, states (and start states), transitions, and tasks explicitly, in basic logical notation"); Garland does not explicitly teach reverse engineering an existing system using the deriving step and the inferring step. However, Bowman-Amuah teaches reverse engineering an existing system using the deriving step and the inferring step (Bowman-Amuah, col. 39, lines 23-40, "… "round-trip" reengineering" provides the developer with a way of modifying a component model and generating the code, then at a later date modifying the code at predefined locations in the source code and regenerating, thus enabling the model to maintain a 2-way-syncronization").

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Garland to include "reverse engineering an existing system using the deriving step and the inferring step" using the teaching of Bowman-Amuah. The modification would be obvious because one of ordinary skill in the art would be motivated to build systems in a development architecture framework to fulfill the requirement (Bowman-Amuah, col. 2, lines 18-23).

Per Claim 25 (Currently Amended):

Garland teaches a method for deriving a process-based specification for a system (Garland, col. 2, lines 25-36, "The method includes accepting, a design specification for the distributed system, including accepting specifications of multiple state machines, and accepting a specification of desired properties of the state machines"); Garland does not explicitly teach reverse engineering an existing system back to a set of traces using the deriving step and the inferring step. However, Bowman-Amuah teaches reverse engineering an existing system back to a set of traces using the deriving step and the inferring step (Bowman-Amuah, col. 39, lines 23-40, "… "round-trip" reengineering" provides the developer with a way of modifying a component model and generating the code, then at a later date modifying the code at predefined locations in the source code and regenerating, thus enabling the model to maintain a 2-way-syncronization").

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the method disclosed by Garland to include "reverse engineering an existing system back to a set of traces using the deriving step and the inferring step" using the teaching of Bowman-Amuah. The modification would be obvious because one of ordinary skill in the art would be motivated to build systems in a development architecture framework to fulfill the requirement (Bowman-Amuah, col. 2, lines 18-23).

Response to Arguments

Applicant's arguments filed 2/4/2009 have been fully considered but they are not persuasive.

Applicant argued:

Basically applicant argued that Carland fails to teach or describe deriving a process-based specification includes deriving a trace-based specification from the trace-base specification from a non-empty set of traces; and mathematically inferring the process-based specification from the trace-based specification, wherein the process-based specification is mathematically equivalent to the trace-based specification, as recited in independent claims 1, 26-28.

Examiner responses:

Garland does teach deriving a trace-based specification from a non-empty set of traces (Garland, col. 2, lines 46-58, “each specification of a state transition includes a specification of values of the state variables in a state in which the transition can be taken, and a specification of an effect on the values of the state variables when the transition is taken”; col. 3, lines 10-15, “deriving a second design specification from the accepted design specification, wherein the derived design specification has a property”; also, col. 22, lines 4-11, "After inputting an initial abstract specification, and after each iteration in which a new specification is derived, the user can determine whether the system specification has the required invariant properties specified by the user…"); **and mathematically inferring the process-based specification from the trace-based specification, wherein the process-based specification is mathematically equivalent to the trace-based specification** (Garland, col. 2, lines 7-15, “The present invention uses a new computer language which is based on a formal, mathematical state-machine model, and which is used both to validate and to generate code for a distributed system. In general, use of this new language enables developing a system using multiple related system specification”, col. 7, lines 29-34, “a new derived specification in the IOA language then becomes part of the user specification”; FIG. 7, col. 13, lines 58-65, “Automaton preprocessor 710 takes the specification of automaton A630, which can make full use of the features of the IOA language, into a from that is usable by the theorem-prover. This form is a mathematical description of the underlying automaton, giving its actions, states (and start states), transitions, and tasks explicitly, in basic logical notation” (emphases added)). Here, Garland teaches deriving a tract-based specification (new/second specification) from a non-empty set of traces (inputting an initial abstract specification/accepted design specification), Garland also teaches mathematically inferring the process-based specification (IOA language) from the trace-based specification (mathematical state-machine model). Thus, Garland teaches all the limitations in claim 1, and other independent claims 26-28.

Applicant argued:

Bowman fails to teach or suggest the claimed subject matter. As to claims 5-7, there is no teaching or suggestion that for at least one input trace, there is one or multiple process expressions output. As to claim 24, Bowman fails to teach or suggest the deriving step or the inferring step. Similarly as to claim 25, Bowman also fails to teach or suggest reverse engineering back to a set of traces.

Examiner responses:

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

As to claims 5-7, Garland teaches at least one input trace, one or multiple process expressions output (Garland, for example, col. 7, lines 25-30, inputs specification text files and produces an output such as text message). As to claim 24, Garland teaches using the deriving step and the inferring step (Garland, col. 2, lines 46-58, “each specification of a state transition includes a specification of values of the state variables in a state in which the transition can be taken, and a specification of an effect on the values of the state variables when the transition is taken”; col. 13, lines 58-65, “Automaton preprocessor 710 takes the specification of automaton A630, which can make full use of the features of the IOA language, into a from that is usable by the theorem-prover. This form is a mathematical description of the underlying automaton, giving its actions, states (and start states), transitions, and tasks explicitly, in basic logical notation”). As to 25, Bowman-Amuah teaches reverse engineering an existing system back to a set of traces using the deriving step and the inferring step (Bowman-Amuah, col. 39, lines 23-40, “… ”round-trip” reengineering” provides the developer with a way of modifying a component model and generating the code, then at a later date modifying the code at predefined locations in the source code and regenerating, thus enabling the model to maintain a 2-way-syncronization”). Thus, the combination of Garland and Bowman teaches all the limitations in claims 5-9, and 24-25.

Conclusion

4. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136 (a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anna Deng whose telephone number is 571-272-5989. The examiner can normally be reached on Monday to Friday 9:30 AM - 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Anna Deng/

Examiner, Art Unit 2191

/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191